# PhoneAsId Iteration Summary

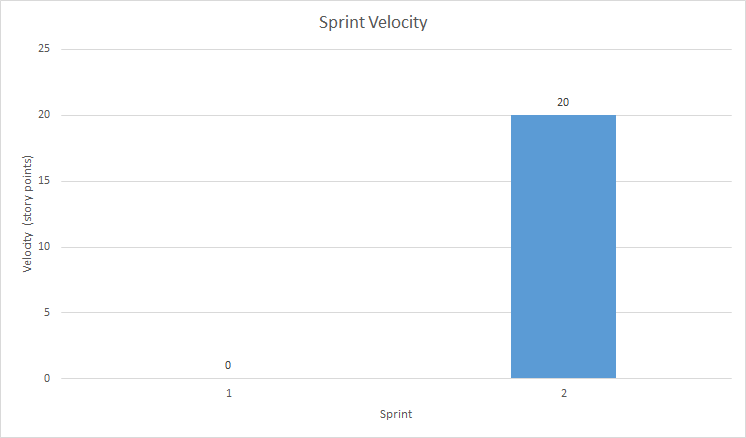
## Team members

|  |  |
| --- | --- |
| **Name** | **Id** |
| Francis Côté-Tremblay | 26615287 |
| Ahmed Dorias | 26649874 |
| Harrison Ianatchkov | 26607403 |
| Sebastian Rafique Proctor-Shah | 26648568 |
| Michal Wozniak | 29649727 |
| Simon Moniere Abes | 21941097 |

## Project summary

This project is meant to virtualize the student ID system at Concordia. A mobile application will be used to allow students to readily have their student IDs with them. It will also allow them to take their own pictures to be used as their picture ID. The picture goes through an initial layer of validation by the mobile application. It is then validated in the backend by a person on a web application. Some future features may include geo-detection with the use of iBeacons, purchasing printer credits, signing up and wirelessly authenticating membership to Concordia’s LeGym, as well as displaying shuttle bus schedules.

## Velocity

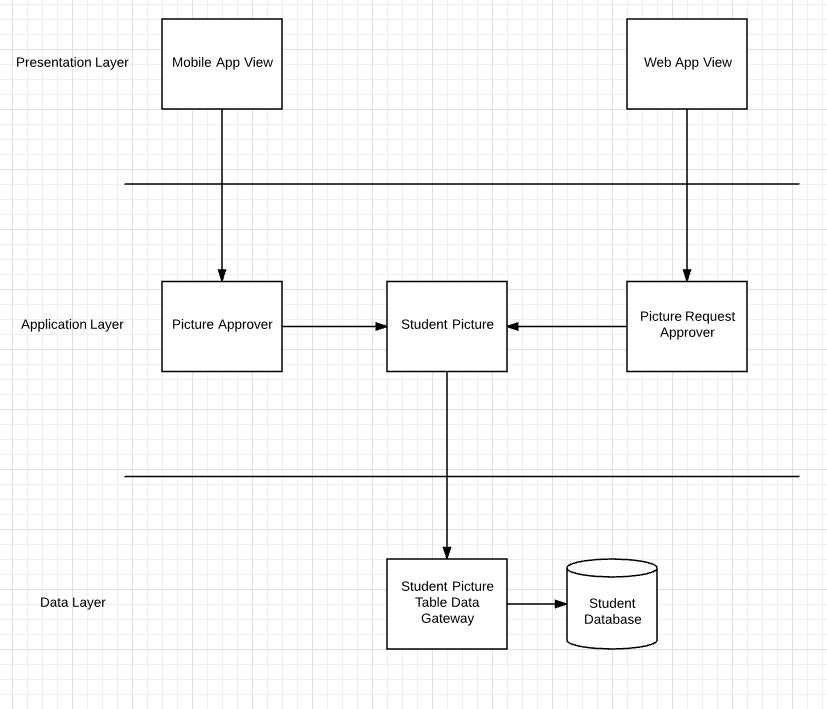


In the first sprint, we were able to complete 0 story points of the 20 estimated. We didn't realize the complexity of integrating oauth2 inside our application. We now considered this issue to be an “Epic Issue”. Our velocity after the first sprint was 0.

In Sprint 2, we were only able to complete 20 story points out of the 28 estimated. For issue “Staff Validation Process” worth 8 points, we had trouble coordinating the structure, layout, and environment of our code. Some of us were working on the front-end of the web application on WebStorm. We were instead supposed to merge front-end development with MVC back-end development in Visual Studio. Other issues included familiarization with AngularJS, CSS, SCSS, npm, and bower technologies. The feature has been built, and it works on WebStorm, but it only works separate from the ASP.NET C# back-end. We therefore consider it incomplete, which results in a total number of story points of 20. For sprint 2, our velocity is 20.

We are still working on issue authentication from sprint 1. We are capable of authenticating using oauth2 with google on the web app. We are missing the ability to authenticate from the mobile application. We will continue to work on it in sprint 3.

## Overall Arch and Class diagram



## The above 3 layered architecture shows that the user interfaces of both applications ( at the presentation layer ) will interact with the Approvers. They will both have access to a Student Picture, in the Application Layer, which will be accessed from the student database through a gateway, both in the Data Layer.

## Plan up to next release

For the first iteration, we will be focusing on allowing the user to successfully be authenticated using their MyConcordia netname and password.

The second iteration will involve features centered around the profile picture. The user will be allowed to take a picture, crop a picture, send a picture, and load a picture previously taken on their phone. We will also set up the picture validation process which will be carried out by the Concordia staff member.

The third iteration will wrap up the features revolving around the profile pictures, in order to prepare the application for its first release. We will work on allowing the user to update the picture at different times during the academic year, as well as setting in-app restrictions for which the profile picture can be submitted by the user.

[Iteration 1](https://github.com/mv740/E-Wok-MyConcordia/milestone/1) (~~20~~ 0 points)

[Iteration 2](https://github.com/mv740/E-Wok-MyConcordia/milestone/2) (~~28~~ ~~48~~ 28 points)

[Iteration 3, Release 1](https://github.com/mv740/E-Wok-MyConcordia/milestone/3) (~~21~~ 41 points, 69 total points)

## Infrastructure

[**Phonegap/ionic**](http://ionicframework.com/)

Using Phonegap was a requirement imposed by our stakeholder, IITS. Their team has experience developing mobile applications using Phonegap, so it will be easier to for them to do maintenance when they eventually take over the project. Phonegap allows for building cross-platform mobile applications that run on iOS, android and windows phones. Ionic is an HTML5 mobile app framework which will improve front end UI development.

[**ASP.NET**](https://www.asp.net/get-started)

ASP.NET was a requirement imposed by our stakeholder. The majority of the concordia’s online system is build using this framework, therefore they would able to continue maintaining the application.

We decided to use the ASP.NET Core version which used to be called ASP.NET 5. Microsoft made this version open-source and added architectural changes which resulted into a cross-platform framework for building modern cloud based and on-premise application. It consists of modular components with minimal overhead, which permit you to keep your flexibility while building your solution.

[**Entity Framework**](https://www.asp.net/entity-framework)

It is an object-relational mapper that enables .Net developers to work with relational data using domain-specific objects. We aren’t required anymore to use raw data-access code.

[**Active Directory**](https://azure.microsoft.com/en-us/documentation/articles/active-directory-aadconnect/)

Active Directory is the directory that will be used to be capable of implementing Oauth 2.0 authorization protocol. This will permit users to login using their concordia netname.

[**Oracle Database (Oracle 12C)**](http://www.oracle.com/us/corporate/features/database-12c/index.html)

For the backend of this application, we will use Oracle Database, more specifically Oracle 12C. Oracle RDBMS is an object-relational database management system. This system is used by IITS in order to store all the information related to the students at Concordia. Therefore, we decided to use it so that the integration with their system will be much easier to do once we hand over the project to IITS developers at the end of our capstone.

[**Amazon Relational database service**](https://aws.amazon.com/rds/)

We are using this service to host our oracle database online. It was the only available service that was offering free hosting for Oracle 12c.

[**Azure**](https://azure.microsoft.com/)

It is a cloud computing platform. We will used to deploy our web application. We can take advantage of Azure’s continuous deployment with github. Whenever we will commit to our Github repo, that code will be automatically synced up with our Azure Web app.

[**Application Insights**](https://azure.microsoft.com/en-us/services/application-insights/)

It is an extensible analytic service that helps you understand the performance and usage of our live web application. It’s designed to help us continuously improve the performance and usability of our application.

## Name Conventions

* [AngularJS styleguide](https://github.com/johnpapa/angular-styleguide/blob/master/a1/README.md)
* [.Net Framework C# Naming Guideline](https://msdn.microsoft.com/en-us/library/ms229002(v=vs.110).aspx)